RZ/N1D Group
RZ/N1S Group
RZ/N1L Group

Application Note:
PROFINET Quick Start Guide

RZ Family   RZ/N1 Series

Preliminary

ARM

Notice

General Precautions in the Handling of the Product

The following usage notes are applicable to all products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Table of Content

Figures

Tables

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

1. Introduction

1 Introduction

This document describes how to run *port GmbH's* PROFINET device stack on the RZ/N1 Series. It is possible either to run a standalone variant only using one core, the CM3-core, or to use two separate cores in the case of the RZ/N1D communicating via Core To Core. Both cores feature the GOAL (Generic OS Abstraction Layer) which handles the communication of the cores and provides basic functionality e.g. timer handling.

The industrial network protocol is located at the CM3-core in both the standalone and the Core To Core variant. Its task is the communication with other operators, therefore the alias name of the CM3-core is communication core (CC) in this document. It contains also a DLR stack that allows to run the RZ/N1 as a Beacon-based Ring Node.

In the Core To Core variant the user application is executed on the Linux based CA7-core. This core is also named as application core (AC). In the standalone variant the user application is running on the communication core, either.

Please note that the software was tested using hardware version
EESS-0401-130-04 (RZ/N1D),
EESS-0401-131-03 (RZ/N1D-EB),
EESS-0401-141-02 (RZ/N1S),
EESS-0401-155-01 (RZ/N1L),
of the CPU and extension board.

Please note that the RZ/N1S requires at least hardware version EESS-0401-141-02 to work with the extension board correctly.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

2. Features

2 Features

The *port GmbH's* PROFINET device stack is a fully implemented Conformance Class A PROFINET stack. It has no feature restrictions but in demonstration mode it only runs about 3 hours. Using the SNMP stack Conformance Class B is also possible.

It contains but is not restricted to the following features:
- Alarms
- Diagnosis
- Cyclic data exchange up to 1 ms cycle time (depending on the application)
- Acyclic data
- Unrestricted record data access (depending on the application)
- Neighbourhood detection support
- DCP support (station name, IP configuration, factory reset)
- LLDP support
- Examples with ready to run projects
- Callbacks for various PROFINET events
- Dynamic module plug and pull support / dynamic device configuration
- Supported by the port GmbH PROFINET I/O Config Tool

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

3   Project Setup

The following chapter describes the setup and usage of the PROFINET device stack.

3.1   Requirements

Please extract the released archive to your workspace and make sure that the following components are installed:

Tool	Version
IAR Embedded Workbench for ARM	8.32.3.20228
IAR C/C++ Compiler for ARM	8.32.3.193
GCC	8.2.0

Table 3-1: Development Tools required by PROFINET device stack

If you need logging, a terminal emulator like putty should be also installed and configured to the correct USB serial interface of the RZ/N1 board. If no messages appear after the board is started than another serial port (from the 4 installed devices) must be tried.

3.2   Hardware

Please take care to follow the setup guidelines for the RZ/N1 Demo Board from the Linux and U-boot documentation - *RZN1x-Quick-Start-Guide.pdf*

Please follow these initial steps to setup the UART and DFU connection.

1. Connect the board to a PC via the UART and the DFU interface. After the driver for the device has been installed, additional serial ports will show up.
    a. On Linux PCs, if you have no other serial-over-USB devices attached, this is */dev/ttyUSB2*.
    b. On Windows PCs, open the *device manager* and look up for new USB Serial Ports on section *ports*. The RZ/N1D and RZ/N1S board uses the 3rd of the 4 COM ports.

2. Open suitable terminal emulator on PC:
    a. On Linux PCs, open a serial terminal e.g. with

        cu -e -o -115200 -l /dev/XXX

        Replace the "XXX" with the serial device where the UART of the board is connected to.

    b. On Windows PCs, open a serial terminal program e.g. PuTTY and select the COM port where the UART is connected to. The following settings must be configured for the connection:

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup
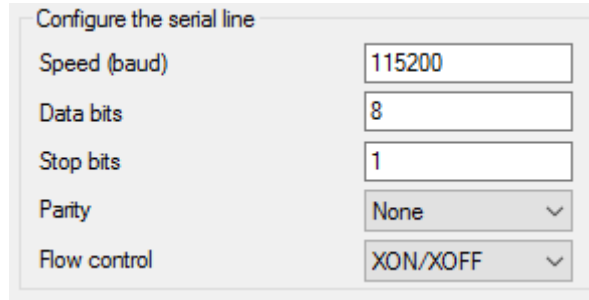
Figure 3-1: Serial Terminal settings RZ/N1

3.3 Sample Application

Several sample applications are provided for *port GmbH*'s PROFINET device stack. They show how to set up and use the stack. The following examples can be found in the folder *goal/appl/goal_pnio/*

00_rpc_cc**	communication core CTC counterpart for AC
01_simple_io	Simple I/O data mirror example
02_io_demo	Maps the I/O data on the LEDs and buttons
03_record_write	Shows how to process a record write
05_ioxs_states	IOPS and IOCS handling
06_apdu_status	Like IOxS status but for APDU
07_alarm_button	Trigger an alarm with a button
08_dynamic_modules	Dynamic module demonstration
09_busy_records	Demo for postponing record handling
10_led_demo	DCP signal on LED demo
11_multiple_write	Process a multiple write request
13_pnio_snmp	Simple I/O data mirror example incl. SNMP
14_info_set	Set the device information data
15_config_set	Set the PROFINET config data
16_device_name*	Set device name in application
17_process_alert*	Cyclically process alarm sending example
18_dyn_mod_postpone*	ExpectedSubmoduleBlock post-poning example
19_subst_val*	Substitute value and APDU status example
20_subst_mod*	Substitute value module acceptance example
21_pnio_snmp_dm***	Simple I/O data mirror example incl. SNMP and MCTC

All applications not marked with * or ** can be used both in the standalone variant on the CC or in the Core To Core variant for the AC.

*   This application is only available on communication core as standalone only.
**   This application is only usable on communication core in Core To Core variant.
***   This application is only available on application core in Core To Core variant.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

3.4   Quick Start for the 01_simple_io application

3.4.1   Application behavior

The application behavior is controlled by the file *appl\goal_pnio\01_simple_io\goal_appl.c*. The first part of the file controls the device configuration (modules and slots), the second part is the application itself that mirrors the received output data to the input data. Input and output directions are always described from the view of the PLC.

3.4.2   Changing MAC Address

The default value of the RZ/N1 board MAC address is 02:00:00:00:00:01.

The MAC address can be set by the application during the initialization of the device by (re)-defining the weak function goal_tgtBoardEthMacGet(). This function is called by the Ethernet driver during its initialization to directly fetch an individual and non-default User-MAC-Address.
Like shown in the following example, the User-MAC-Address must only be copied to the address of the corresponding function parameter:

```
/** Default MAC address read function
 *
 * @retval GOAL_OK successful
 * @retval other failed
 */
GOAL_STATUS_T goal_tgtBoardEthMacGet(
    uint8_t *pMacAddr                              /**< pointer to driver MAC address */
)
{
    /* declare and initialize individual user device MAC address */
    GOAL_ETH_MAC_ADDR_T userDeviceMac = { 0x02, 0x00, 0x00, 0x00, 0x00, 0xaa };

    /* copy user MAC address to MAC address in Ethernet driver */
    GOAL_MEMCPY(pMacAddr, userDeviceMac, MAC_ADDR_LEN);

    return GOAL_OK;
}
```

Figure 3-2: Code example for setting User device MAC-Address

3.4.3   Changing IP Address

The IP address is managed by the PLC. With a DCP configuration tool like "Tia Portal" or "Primary Setup Tool" the IP address can be changed.

3.5   Running the *01_simple_io* application

The RZ/N1D and RZ/N1S use the U-Boot bootloader for initial setup of the hardware and loading of the CM3 firmware. Additionally, the RZ/N1D U-Boot bootloader is used for booting the Linux Kernel. The RZ/N1L is working without any bootloader. This chapter describes how to install the management software on the flash of the board. If no bootloader was yet installed on the RZ/N1D and RZ/N1S please

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

refer to the Linux documentation - Quick Start Guide for U-Boot and Linux - *RZN1x-Quick-Start-Guide.pdf.*

There are many similarities between the derivatives of the RZ/N1 series but some minor differences, too. Therefore here is a more detailed explanation how to run a sample application on each.

All standalone projects and the CC project of the Core To Core variant contain different workspaces for each board variant. The project workspaces ending on *_eb contain the configuration for the CPU Board together with the extension board (4 switch ports). The other project workspaces contain the configuration for working with the CPU Board only.

3.5.1   Standalone Variant – RZ/N1D and RZ/N1S

It is possible to load the code via debugger into RAM, which is a very fast approach to test the user application, or to flash the CM3 core. In both cases any application located in *goal\projects\goal_pnio_lib* and must be built using IAR Embedded Workbench.

3.5.1.1. Loading application into RAM via IAR Embedded Workbench

To compile a project, follow these steps:

1.   Start the IAR Workbench IDE
2.   Open a project via "File/Open Workspace".
3.   Go to the workspace folder and open it. In case the CPU board is used together with the extension board, please ensure to select the correct IAR-project.
4.   Compile the project via "Project/Compile" or "Project/Rebuild all".
5.   Power up the device
6.   Open a serial terminal according to section 3.2.
7.   Press any key on your keyboard to interrupt the bootloader.
8.   Ensure to configure the U-Boot boot command to release the CM3 core after reset. This is done by the command:

   setenv bootcmd "mw 0x04000004 1 && rzn1_start_cm3 && loop 0 1"

   followed by

   saveenv

   and reset the board.
9.   Connect the debugger to the system via the "Download and Debug" button of the IAR Workbench.
10. After the Debug view opened, click on the "Go" button.

3.5.1.2. Loading application into flash via dfu-util

The board uses the U-Boot bootloader for initial setup of the hardware and loading of the CM3 core firmware. This chapter describes how to install the compiled management software on the flash of the board. If no bootloader was yet installed on the board, please refer to the Linux documentation - Quick Start Guide for U-Boot and Linux - *RZN1x-Quick-Start-Guide.pdf.*
The following steps describe the installation of management software:

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

1. Connect a Linux PC to the board according to section 3.2.
2. Power up the board.
3. Open a serial terminal according to section 3.2.
4. Hit any key to stop the autoboot of the U-Boot
5. Type "*dfu*〃 in the serial terminal of the board and hit enter.

6. On a Linux terminal start the command

    sudo dfu-util -a "sf_cm3" -D FIRMWARE.bin

    Replace *FIRMWARE.bin* with the file name of the software to install. The binary is placed at the
    subfolder *Debug-RAM\Exe* of the IAR project folder.
7. When the download process is complete, press Ctrl+C on U-Boot.
8. If the autoboot command was already configured, go to step 10.
9. Set the autoboot command in the U-Boot:

    setenv bootcmd "sf probe && sf read 0x4000000 d0000 90000 && rzn1_start_cm3 && loop 0 1"

10. Save the command to the flash:

    saveenv

11. Reset the device

3.5.2. Standalone Variant – RZ/N1L

The RZ/N1L does not use any bootloaders. If any application is stored in flash, it will be started
automatically. Both, loading into RAM and flash can be done using IAR workbench.

1. Start the IAR Workbench IDE.
2. Open a project via "File/Open Workspace".
3. Go to the workspace folder and open it.
4. Compile the project via "Project/Compile" or "Project/Rebuild all".
5. Power up the device.
6. Open a serial terminal according to section 3.2.
7. Choose either the Debug-RAM or the Debug-ROM configuration. First is used for debugging via
    IAR, second is loading the application into the flash.
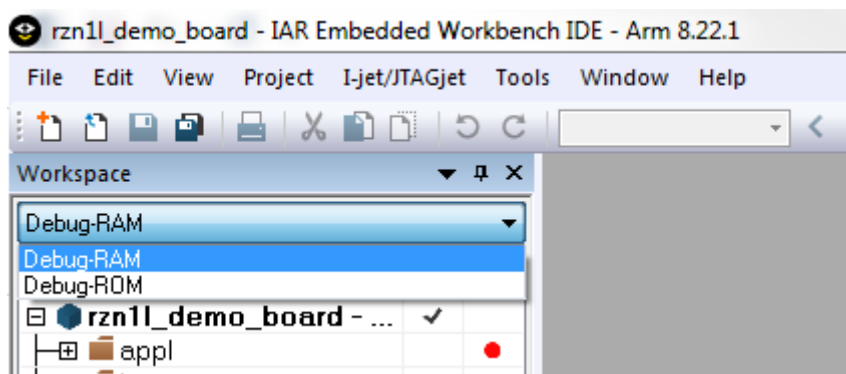
Figure 3-3: IAR Configurations RAM and ROM for RZ/N1L

8. Follow these steps for the Debug-RAM configuration

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

    a. Compile the project via "Project/Compile" or "Project/Rebuild all".

    b. Press and hold the devices software-reset button.

    c. Click on "Download and Debug" and release the software-reset button as soon as the "Busy" window opens.
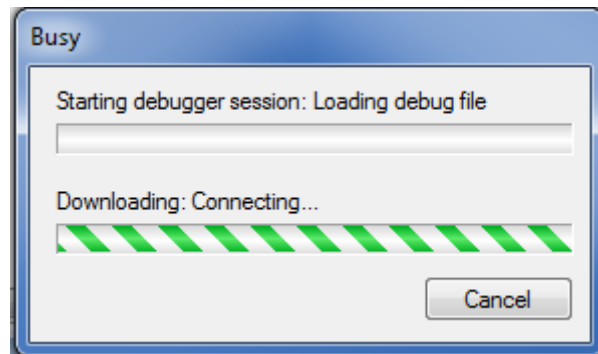
Figure 3-4: IAR Workbench "Busy"-Window

9. Follow these steps for the Debug-ROM configuration
   a. Click on "Download and Debug".
   b. Set reset mode to "Hardware" and press "Make & Restart Debugger".
   c. Check, if the reset mode is still on "Hardware". If not, repeat the previous step.
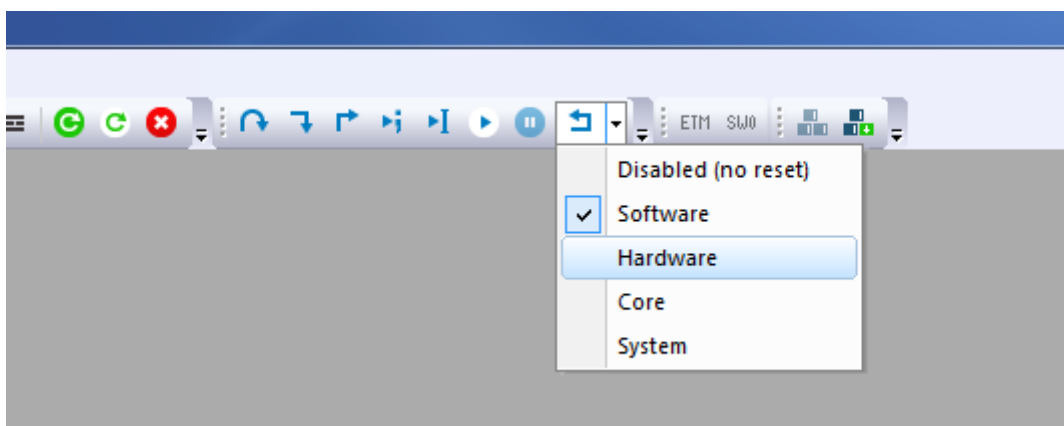
Figure 3-5: Changing Reset mode of RZ/N1L in Debug-ROM configuration

After the Debug view opened, click on the "Go" button.

3.5.3 Core To Core variant – RZ/N1D (Communication Core)

The binary file for the CM3 core is located at the board type related IAR Embedded Workbench folder *goal\projects\goal_pnio_rpc\00_goal_rpc* respectively
goal\projects\goal_pnio_rpc\00_goal_rpc_demo.

Load the binary file to the flash according to the following steps.
1. Connect a Linux PC to the board according to section 3.2.
2. Power up the board.
3. Open a serial terminal according to section 3.2.
4. Hit any key to stop the autoboot of the U-Boot
5. Type "*dfu*" in the serial terminal of the board and hit enter.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

6. On a Linux terminal start the command

   sudo dfu-util -a "sf_cm3" -D FIRMWARE

   Replace *FIRMWARE* with the file name of the software to install.
7. When the download process is complete, press Ctrl+C on U-Boot.
8. If the autoboot command was already configured, go to step 10.
9. Set the autoboot command in the U-Boot:

   setenv bootcmd "sf probe && sf read 0x4000000 d0000 90000 && sf read 0x8ffe0000 b0000 20000 && sf read 0x80008000 1d0000 f00000 && rzn1_start_cm3 && sleep 4 && bootm 0x80008000 - 0x8ffe0000"

10. Save the command to the flash: *saveenv*
11. Reset the device

It is also possible to debug the RZ/N1D communication core. The steps accored to section 3.5.1.1, but the boot command has to be set to

   setenv bootcmd "mw 0x04000004 1 && rzn1_start_cm3 && sleep 20 && sf probe && sf read 0x80008000 1d0000 f00000 && sf read 0x8ffe0000 b0000 20000 && bootm 0x80008000 - 0x8ffe0000"

This delays the boot of Linux about 20 seconds. Meanwhile the CM3 core has to be startet.

3.5.4 Core To Core variant – RZ/N1D (Application Core)

The user application runs on the Linux system of the CA7. Its binary must be created by GCC and downloaded to the RZ/N board manually.

> *Note: It is recommended to maintain a consistent boot order of the communication and the application core. Therefore it is advised to always start the R-IN engine / Cortex-M3 (communication core) first and boot the Cortex-A7 (application core) after the communication core has finished its initialization.*

3.5.4.1 Building and downloading the user application

The following steps describe, how to build a binary and download it to the RZ/N1 board.

1. Navigate with the terminal of a Linux PC to the project of the application core at *goal/projects/goal_pnio_rpc_lib/01_simple_io/gcc.*
2. Start the build process by executing the Makefile by typing
     make

3. Select as target platform "rzn_a7_demo_board".
4. Power up the board and wait till Linux booted successfully.
5. Copy the created binary file *build/rzn_a7_demo_board/goal_rzn_a7_demo_board.bin* to the RZ/N1 board by e.g. secure copy (scp).
6. Copy the application corresponding library *goal/projects/goal_pnio_rpc_lib/00_lib/gcc/build/rzn_a7_demo_board/libgoal_rzn_a7_demo_board.so* to the RZ/N1 board by e.g. secure copy (scp).
7. Start the binary file on the target by typing the commands
         export LD_LIBRARY_PATH=.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

./goal_rzn_a7_demo_board.bin –i eth0

The GOAL setups the connection to the communication core via core to core and starts the user application. The initialization is done when the log message *"GOAL initialized"* is printed at the terminal, if logging is activated.

3.5.4.2   Auto start the user application

The Linux Kernel can start the user application on the CA7 automatically with the help of the start script

S99goal_app.sh

This script is placed at *linux_ctc/* of the release. Download the file to the CA7, like the user application binary, and place it at *etc/rc5.d/ if this file is not present*. Please ensure, that *goal_rzn_a7_demo_board.bin* and its library is placed at */home/root/*.

Disabling the start script is possible by adding the boot argument *GOAL_APPL_LINUX_PREV*.

1. Power up the board.
2. Hit any key to stop the autoboot of the U-Boot
3. Add the boot argument for preventing the application autoboot by

    setenv bootargs "${bootargs} GOAL_APPL_LINUX_PREV"

4. Save the command to the flash by:
    saveenv

5. Reset the device

Reenabling the start script is possible by deleting the boot argument *GOAL_APPL_LINUX_PREV.*

1. Power up the board.
2. Hit any key to stop the autoboot of the U-Boot
3. Display the environment by

    env print

4. The latest boot arguments are listed at the line *bootargs=*
5. Copy these arguments, except *GOAL_APPL_LINUX_PREV* and paste them at <paste> on the following command

    setenv bootargs "<paste>"

3.5.5. 6. Save the command to the flash by:

    saveenv

7. Reset the device

Core To Core variant – RZ/N1S

Similar to the standalone variant the Core To Core variant on the RZ/N1S is also capable to run from the RAM while debugging the application core and the communication core at the same time.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

The IAR Embedded Workbench runs two instances of the IDE, one for each core, in a master-slave-system to share the access to the board keeping both instances synchronous.

The usage and setup of the multicore debugging will be exemplary described for the Simple IO example running on the application core under ThreadX using the CM3 for handling the PROFINET device stack as the communication core.

> *Note:* *It is recommended to maintain a consistent boot order of the communication and the application core. Therefore it is advised to always start the R-IN engine / Cortex-M3 (communication core) first and boot the Cortex-A7 (application core) after the communication core has finished its initialization.*

To run the core to core variant of the Simple IO example please perform the following steps:

1. Power up the board.
2. Open the corresponding AC IAR project workspace, e.g.:

   projects\goal_pnio_rpc\01_simple_io\iar\renesas\rzn1s_a7_threadx\rzn1s_a7_threadx.eww

3. Open the project options and navigate to the subcategory "Multicore" in the category "Debugger".
4. Enable Multicore master mode and select the slave workspace to use. Please note, that the "slave project" and the "slave configuration" is already preconfigured for the GOAL slave projects.
   The core to core variant requires the *00_goal_rpc_demo* project running on the CM3 which is the same project as for the core to core variant under Linux on the RZ/N1-D but for the RZ/N1-S demo board. This slave application can be used also for the other PROFINET application core demo applications.
   The slave workspace *rzn1s_demo_board.eww* is located in the following project directory:

   projects\goal_pnio_rpc\00_goal_rpc_demo\iar\renesas\rzn1s_demo_board
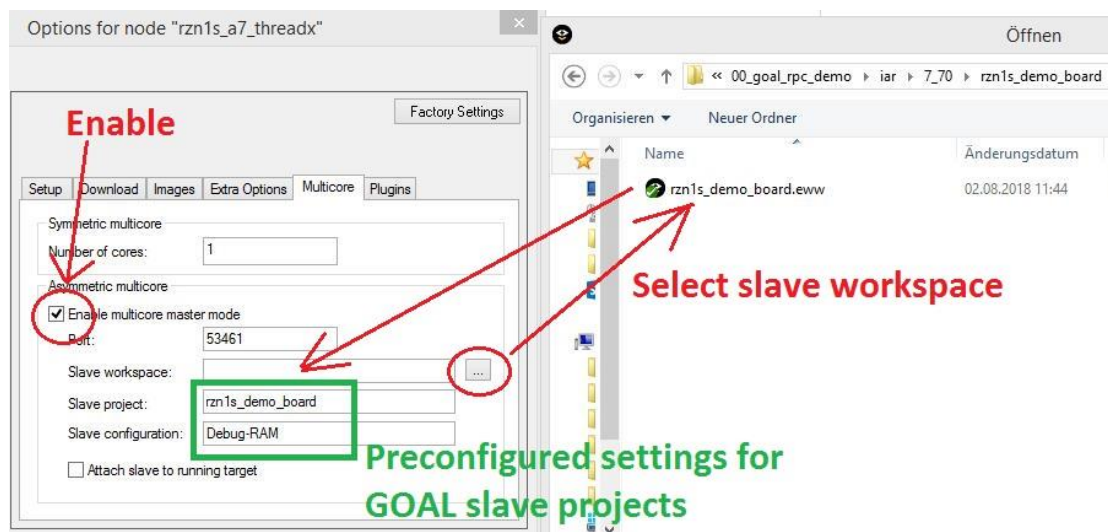
Figure 3-6: Multicore Debug Option

When using the RZ/N1S expansion board, please ensure to select the correct CC project located at the projects rzn1s_demo_board_eb directory. Additionally, adjust the entry "Slave project"

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup

in the subcategory "Multicore" to *rzn1s_demo_board_eb*.

5. Compile the project via "Project/Compile" or "Project/Rebuild all".
6. Press the "Download and debug" button or Ctrl+D
   This will cause IAR to open the slave workspace as an additional IAR workbench instance, build the slave project and load both – the master and the slave project – to the board sharing the debugger.
7. When the software from both instances is loaded to the board and the IDE switches in the debug mode an additional dialog for multicore debugging is available giving the following options:

   - start all cores at once
   - stop all cores at once
   - toggle execution mode
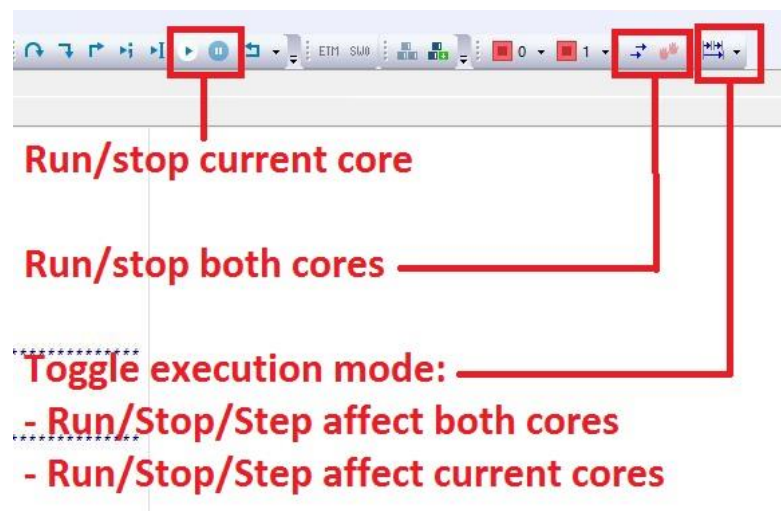
Figure 3-7: Multicore Debug Interface

3.5.6.

Core To Core variant – RZ/N1L (Communication Core)

The binary file for the CM3 core is located in the board type related IAR Embedded Workbench folder *goal\projects\goal_pnio_rpc\00_goal_rpc* respectively *goal\projects\goal_pnio_rpc\00_goal_rpc_demo*.

Please refer section 3.5.2 for building and downloading the Core to Core variant on RZ/N1L. It is handled the same as the standalone variant.

For mulit core projects, the RZ/N1L is used as communication core, while the e.g. Synergy S7GS-SK is used as application core. Data exchanging is done by SPI. The boards are connected as followed.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

3. Project Setup
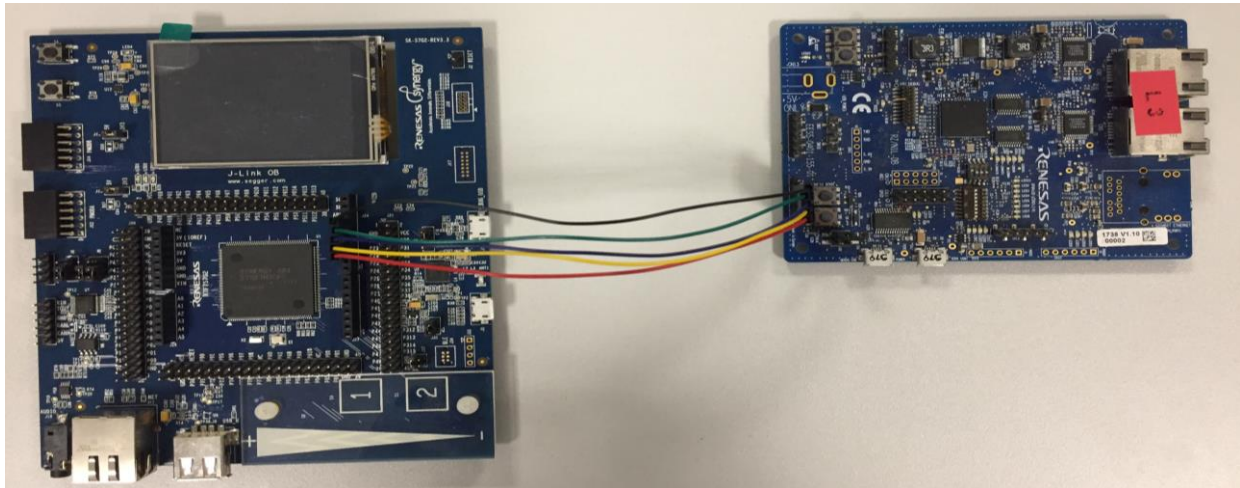
Figure 3-8: SPI connection of Synergy S7GS-SK (left) and RZ/N1L (right)

function	color	S7GS-SK	RZ/N1L
GND	Black	J24-7	CN20-5
SPI Clock	Green	J24-6	CN20-4
MISO	Blue	J24-5	CN20-3
MOSI	Yellow	J24-4	CN20-2
SPI chip select	Red	J24-3	CN20-1

Table 3-2: PINs for SPI usage

Please note the synergy quick start guide for setup the named core. By default, the RZ/N1L uses the SPI channel 5 and the following GPIOs

GPIO	Usage
62	SPI clock
63	MOSI
64	MISO
65	SPI chip select

Table 3-3: GPIOs for SPI usage

Note:

> **The board supports only SPI mode 1 and 3. Please set the SPI mode to 3 by defining GOAL_GLOB_MA_SPI_ID_0_MODE_3 in goal/goal_global/goal_global.h to 1.**
>
> *#define GOAL_GLOB_MA_SPI_ID_0_MODE_3 1   /**< set SPI mode 3 on MA ID 0 */*

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

4. Setting up a PROFINET
PLC (Tia Portal with S7-
1500)

4. Setting up a PROFINET PLC (Tia Portal with S7-1500)

This chapter shows how to setup a PROFINET PLC by using the Siemens tool TIA ("Totally Integrated Automation" – Version V13 SP1 Update 4). The step by step instructions are exemplarily done for the PLC "SIMATIC 1511-1 PN".

4.1. Preparation

1. Connect the devices (RZ/N1 to PLC, PLC to PC with TIA)
2. Change the IPv4 address of the computer interface to a private network address (e.g. 192.168.0.30)

4.2. Setting up PLC

1. Open the Siemens tool TIA and create a new project
2. Switch to the "Project view"
3. Load the GSDML file (Menu Options > Manage GSD) and install it
   o The GSDML file can be found in *goal\protos\pnio\gsdml*
4. Add the RZ/N device to the project (Figure 4-1):
   o Double click on "Devices & networks"
   o Open "Network view"
   o Drag & drop the device "Standard" from catalogue directory (Other field devices > PROFINET IO > Drives > I/O > port GmbH > port GmbH > port IO Examples)
   o There should be 4 devices in the listing. The 2 port DAP must be used for the standalone RZ/N1 board and the 4 port DAP for the extension board.
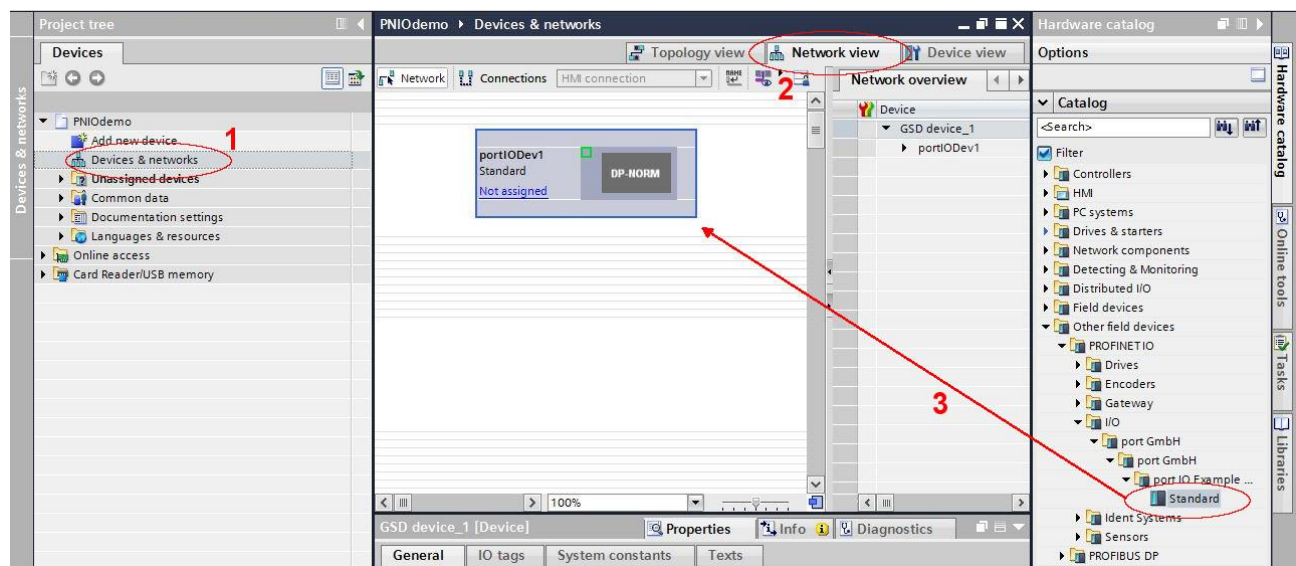
Figure 4-1: Add device to project

5. Add input and output modules to RZ/N1 (Figure 4-2):
   o Change to "Device view"
   o Drag & drop the modules "64 bytes I" and "64 bytes O" from hardware catalogue to the device

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

**4. Setting up a PROFINET
PLC (Tia Portal with S7-
1500)**

Figure 4-2: Add I/O modules

6. Add the PLC to the project (Figure 4-3).
   o Double click on "Add new device".
   o Choose the device and the firmware version 1.7
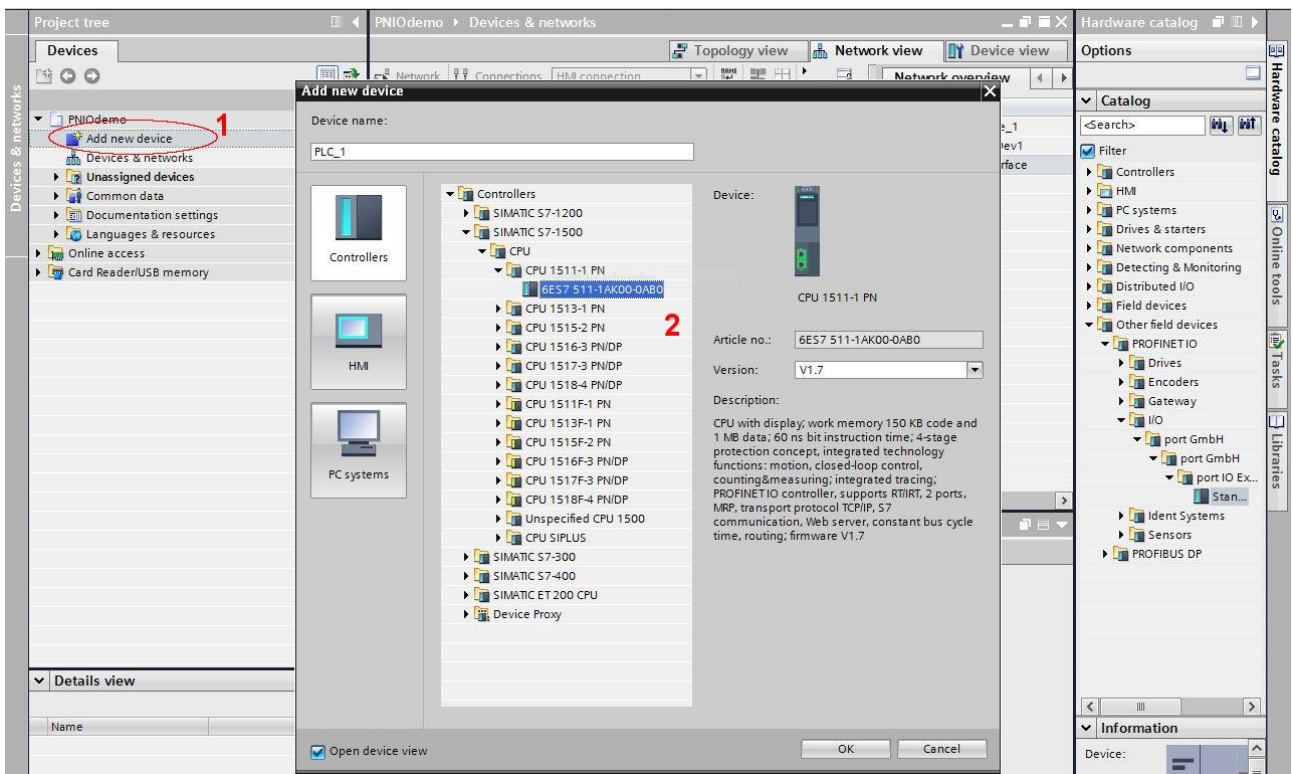
Figure 4-3: Add the PLC to the project

7. Connect the devices (Figure 4-4):
   o Change to "Network view"
   o Connect the green marked Ethernet ports of the devices (drag & drop)

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

4. Setting up a PROFINET
PLC (Tia Portal with S7-
1500)
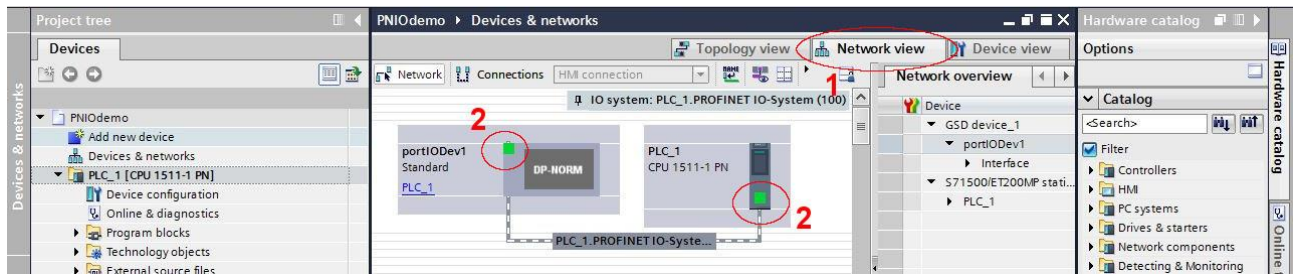
Figure 4-4: Connect the devices

8. Check the IP address of the PLC (Figure 4-5):
   o Change to "Device view"
   o Choose the PLC device
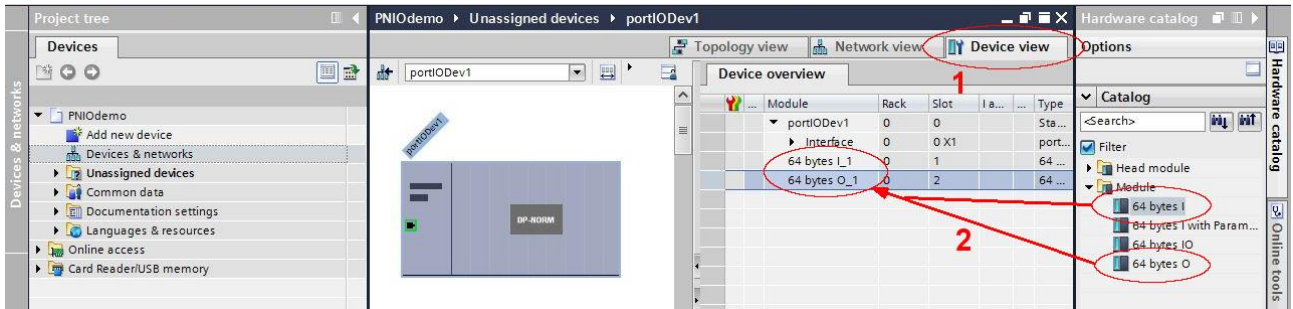   o Click on the PLC in the "Device view"
   o The IP address in the "General" window must be the same as set in the PLC
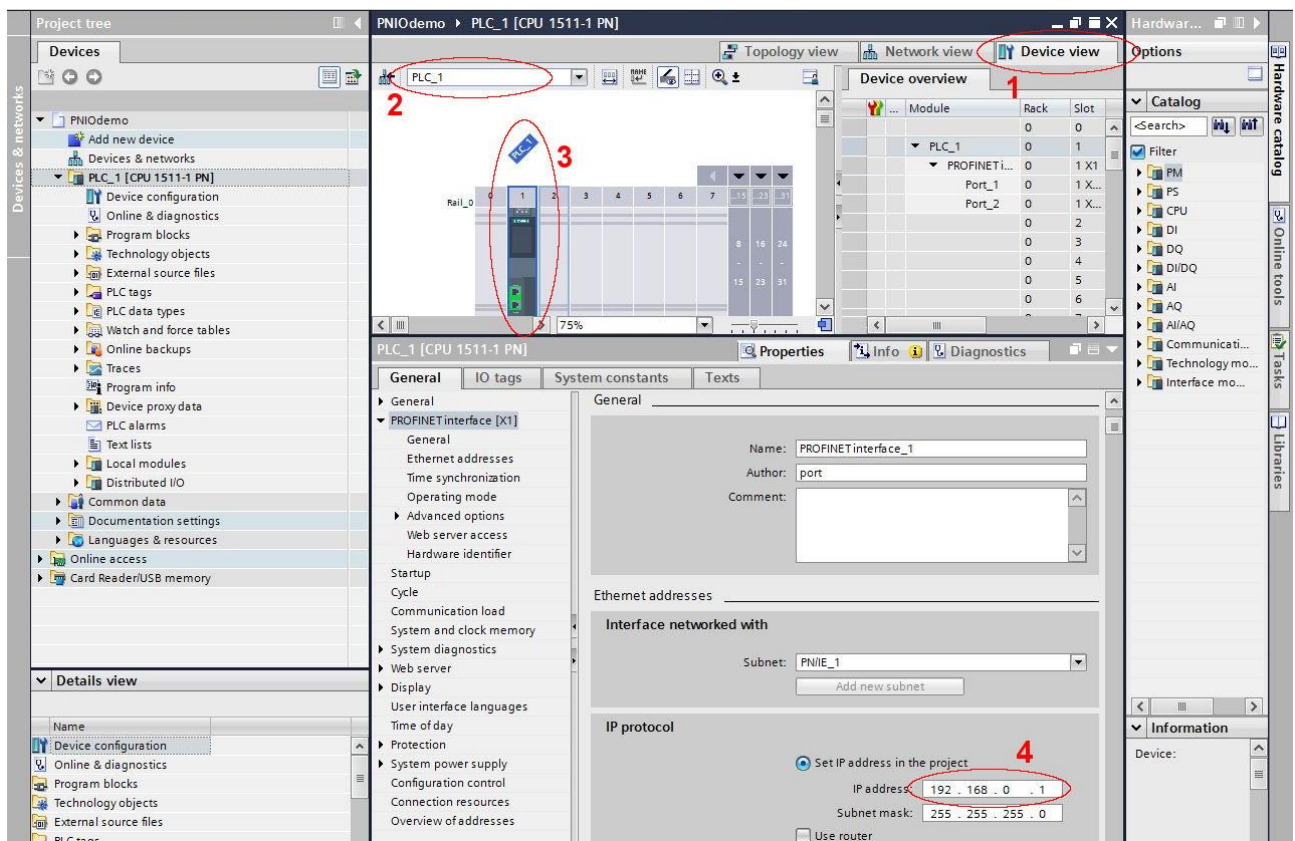
Figure 4-5: Check the IP address

9. Set the name of RZ/N1 (Figure 4-6):
   o Double click on your network interface card (in "Project tree" > Online access)
   o Open "Online & diagnostics" of the RZ/N1 device
   o Go to the function setting "Assign Name"
   o Change the device name
   o Press "Assign name" button

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

**4. Setting up a PROFINET
PLC (Tia Portal with S7-
1500)**
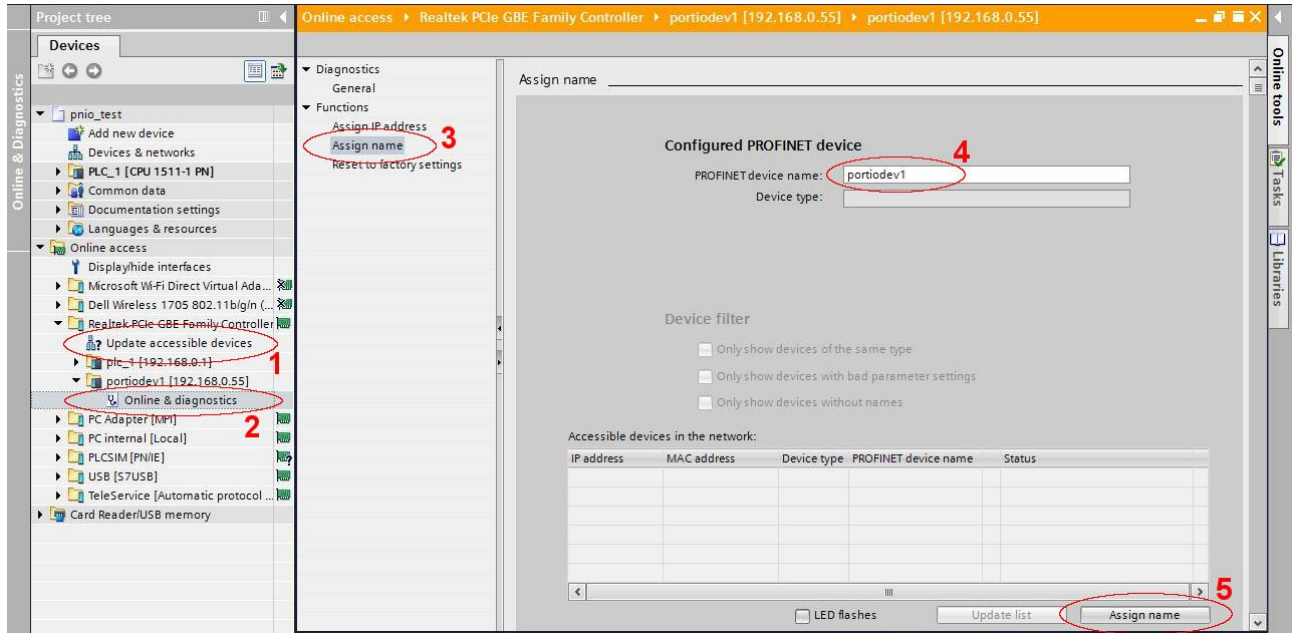
Figure 4-6: Set device name

10. Load the settings to the hardware:
    o Right-click on the PLC at the "Project tree" (left hand side)
    o In the right-click context menu: Download to device > Hardware configuration

    Popup of a new window (Figure 4-7):
    o Select your interfaces and connection
    o Click "Start search" to find your device
    o Select the PLC line
    o Press "Load"

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

**4. Setting up a PROFINET
PLC (Tia Portal with S7-
1500)**
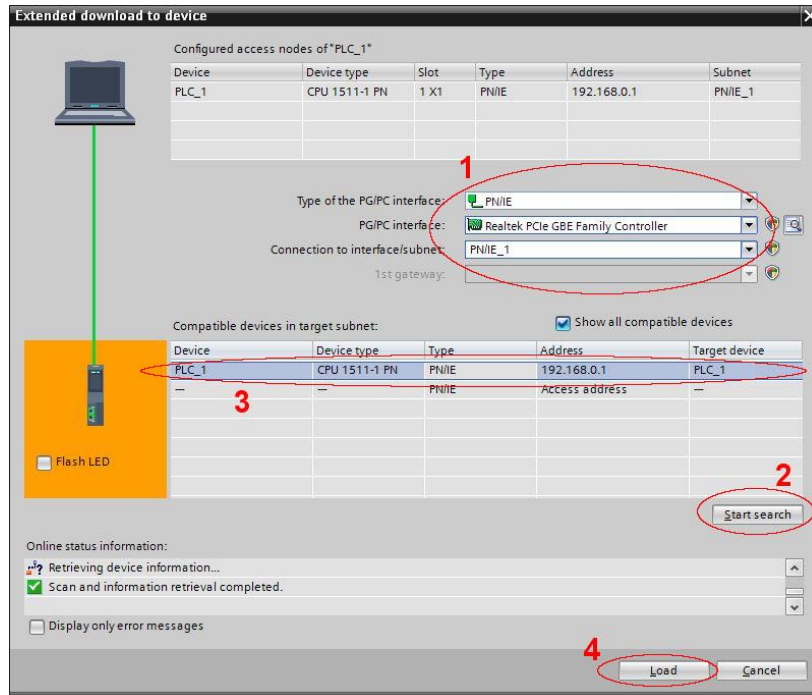
Figure 4-7: Configuration of the connection

Finish PLC configuration (Figure 4-8)
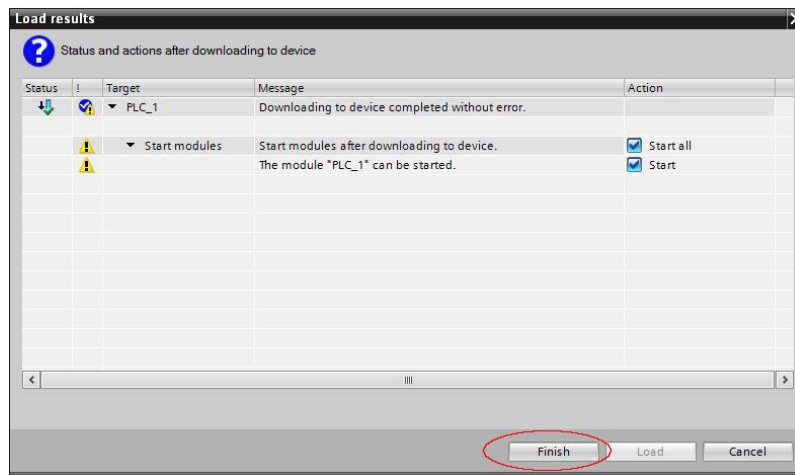
Figure 4-8: Finish configuration

Now the PLC should connect to the device and exchange cyclic data with it. As the data is mirrored in
the *01_simple_io example* the PLC can be also programmed to send cyclic over the output module
which should be mirrored on the input module.

Watching and changing the cyclic data can also be easily achieved with the port GmbH PROFINET I/O
Config Tool.

RZ/N1D/S/L Group
Application Note
PROFINET Quick Start Guide

Preliminary document. Specifications in this
document are tentative and subject to change.

5. Revision History

5. Revision History

Revision History

| Version	Process		Check		Release	
	Date	Name	Date	Name	Date	Name
1.0	2017-04-03	Sven Bachmann	2017-04-03	Marcus Züche	2017-04-03	Sven Bachmann
Initial version.						
1.1	2017-05-04	Marcus Züche	2017-05-04	Marcus Tangermann	2017-05-04	Marcus Tangermann
Update description for U-Boot 2017.01. Change file path by document name. Minor text updates. Remove section Communication core in debug mode						
1.2	2017-07-31	Marcus Züche				
Minor text updates						
1.3	2017-08-07	Marcus Züche				
Review Renesas						
1.4	2017-12-15	Marcus Züche				
Add description for autostart the CA7 application. Minor updates.						
1.5	2018-05-28	Martin Herberg				
Add description project variants and all RZ/N1 derivatives. Minor Updates						
1.6	2018-07-04	Martin Ehlert				
Added interface usb0 as parameter to start command in chapter 3.5.4.1						
1.7	2018-06-22	Marcus Züche	24.08.2018	Martin Ehlert		
Added RZ/N1S CTC and RZ/N1L CTC variant, summary hardware initialization and minor text updates/formation.						
1.8	2019-01-07	Marcus Züche				
Expand description of RZ/N1L SPI RZ/N1S-EB configuration. Add GCC and board versions. Update boot commands.						
1.9	2019-01-23	Martin Ehlert				
Updated description for setting user MAC address and addad advices for Core-to-core boot order						
1.4.3 (1.10)	2019-07-09	Martin Ehlert	2019-07-12	Marcus Züche		
Updated project and application list.						

RENESAS

Renesas Electronics Corporation

Sales Offices

www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RZ/N1 Group

RENESAS

Renesas Electronics Corporation